

1. a. What is a virtual machine?

La idea de una máquina virtual puede ser percibida por los desarrolladores como "una computadora dentro de una computadora" o "una computadora simulada en software". Aunque se pueda imaginar el bytecode como "código máquina para la CPU de la computadora interna" o "código máquina para un procesador inventado", esta percepción simple puede ser engañosa.

b.

Diferencias	Java	Python	JavaScript
Tipo de lenguaje	Es un lenguaje estáticamente tipado, lo que significa que los tipos de datos de las variables se verifican en tiempo de compilación.	Es un lenguaje dinámicamente tipado, lo que significa que los tipos de datos de las variables se verifican en tiempo de ejecución.	Al igual que Python, es un lenguaje dinámicamente tipado.
Sintaxis	Tiene una sintaxis más verbose y está orientado a objetos.	Tiene una sintaxis más concisa y fácil de leer, con un enfoque en la legibilidad del código.	Tiene una sintaxis flexible y está orientado a objetos.
Uso principal	Se utiliza principalmente en el desarrollo de aplicaciones empresariales y móviles.	Es popular en el desarrollo web, scripting, análisis de datos y desarrollo de aplicaciones de inteligencia artificial.	Se utiliza para el desarrollo web y para crear aplicaciones del lado del cliente en navegadores web.
Tipos de datos	Utiliza un sistema de tipos fuertemente tipado.	Es dinámicamente tipado, lo que significa que las variables pueden tomar diferentes tipos de datos durante la ejecución del programa.	También es dinámicamente tipado, lo que permite una mayor flexibilidad en el manejo de tipos de datos.
Promedio de descargas	Se estima que hay alrededor de 9 millones de desarrolladores que utilizan Java en todo el mundo.	Se estima que hay alrededor de 8 millones de desarrolladores que utilizan Python en todo el mundo.	Se estima que hay alrededor de 11 millones de desarrolladores que utilizan JavaScript en todo el mundo.

c. 1. Comentario en Java:

Para poner un comentario en java se deben utilizar los `//`, o un comentario de varias líneas utilizando `/*` para iniciar el comentario y `*/` para finalizarlo.

2. Palabra reservada:

Una palabra reservada en Java es una palabra que tiene un significado especial y está reservada para su uso en el lenguaje. Estas palabras no pueden ser utilizadas como identificadores (nombres de variables, clases, métodos, etc.) en el código. Algunos ejemplos de palabras reservadas en Java incluyen ``public``, ``class``, ``static``, ``if``, ``else``, ``while``, ``for``, ``return``, entre otras.

3. Identificador:

Un identificador es el nombre que se le da a una variable, método, clase u otro elemento del programa. Los identificadores deben comenzar con una letra, un guion bajo (`_`) o un signo de dólar (`$`), seguido de letras, dígitos u otros guiones bajos. Los identificadores distinguen entre mayúsculas y minúsculas. Ejemplo: `int miVariable = 10;`

4. Literal:

En Java, un literal es una representación directa de un valor, como un número, una cadena de texto, un carácter, etc. Por ejemplo, `10` es un literal entero, `"Hola"` es un literal de cadena de texto, y `'a'` es un literal de carácter.

3. En Java, los datos primitivos son tipos de datos fundamentales que representan valores simples. Estos tipos de datos están predefinidos en el lenguaje y no son instancias de clases. Los datos primitivos en Java son: ``int``, ``double``, ``boolean``, ``char``, ``byte``, ``short``, y ``long``.

9. La sentencia "switch" en Java es una estructura de control que se utiliza para tomar decisiones basadas en el valor de una expresión. Funciona de la siguiente manera:

- La expresión dentro del switch es evaluada una vez y su valor se compara con los casos (case) dentro de la estructura.
- Si el valor de la expresión coincide con el valor de un caso (case), se ejecuta el bloque de código asociado a ese caso.
- Si no hay coincidencias, se puede proporcionar un caso predeterminado (default) que se ejecutará si ningún caso coincide con el valor de la expresión.
- Después de ejecutar el bloque de código de un caso, la ejecución continúa hasta encontrar una instrucción "break" o el final del switch.
- La instrucción "break" se utiliza para salir del switch y evitar la ejecución de los casos restantes.

12. ``break`` se utiliza para salir por completo del ciclo en el que se encuentra, mientras que ``continue`` se utiliza para saltar la iteración actual y pasar a la siguiente iteración del ciclo sin salir por completo del ciclo. Ambas sentencias son útiles para controlar el flujo de ejecución en un ciclo según ciertas condiciones.